

hs2865

Hiroaki Suzuki (hs2865)

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.3

library(ggmap)

## Warning: package 'ggmap' was built under R version 3.2.3

master = read.csv("GlobalFloodsRecordMaster.csv", as.is = TRUE)
master_f = data.frame(master)

#### information of USA

## getting info of usa
usa_master = master_f[master_f$Country == "USA", ]

## getting map of usa
map_usa <- get_map(location = "usa", maptype = "satellite", zoom = 4)

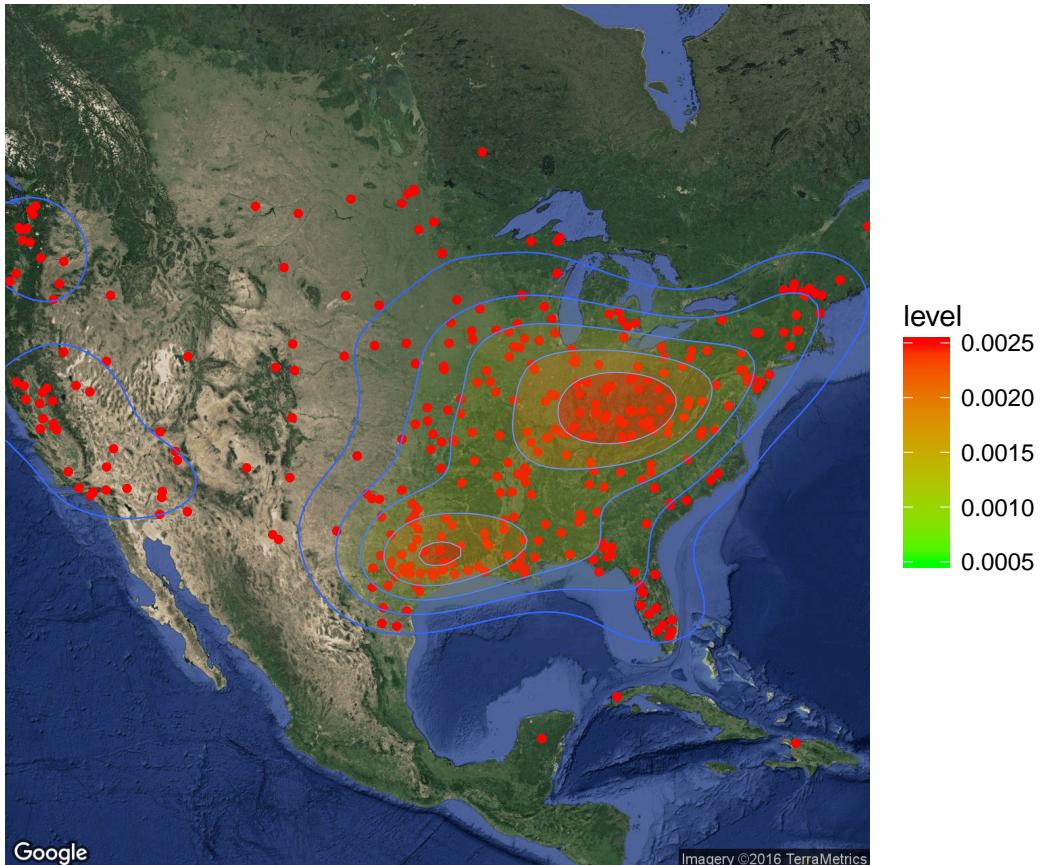
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=usa&zoom=4&size=640x640&scale=2&
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=usa&sensor=false

## Showing which area had floods in the past
usa_master_heat = data.frame(Centroid.X=as.numeric(usa_master$Centroid.X),
                               Centroid.Y=as.numeric(usa_master$Centroid.Y),
                               Total.floods.M.4=usa_master$Total.floods.M.4)

ggmap(map_usa, extent = "device") +
  geom_point(aes(x=usa_master_heat$Centroid.X,
                 y=usa_master_heat$Centroid.Y),
             data=usa_master_heat, col="red", size=1) +
  geom_density2d(data=usa_master_heat,
                 aes(x = usa_master_heat$Centroid.X,
                     y = usa_master_heat$Centroid.Y),
                 size = 0.3) +
  stat_density2d(data=usa_master_heat,
                 aes(x = usa_master_heat$Centroid.X,
                     y = usa_master_heat$Centroid.Y,
                     fill = ..level..,
                     alpha = ..level..),
                 size = 0.01, geom = "polygon") +
  scale_fill_gradient(low = "green", high = "red") +
  scale_alpha(range = c(0, 0.3), guide = FALSE)

## Warning: Removed 46 rows containing non-finite values (stat_density2d).
```

```
## Warning: Removed 46 rows containing non-finite values (stat_density2d).
## Warning: Removed 46 rows containing missing values (geom_point).
```



```
##### information of China
## getting info of china
china_master = master_f[master_f$Country == "China", ]

## getting map of china
map_china <- get_map(location = "china", maptype = "satellite", zoom = 4)

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=china&zoom=4&size=640x640&scale=1
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=china&sensor=false

## Showing which area had floods in the past
china_master_heat = data.frame(Centroid.X=as.numeric(china_master$Centroid.X),
                                 Centroid.Y=as.numeric(china_master$Centroid.Y),
                                 Total.floods.M.4=china_master$Total.floods.M.4)

ggmap(map_china, extent = "device") +
  geom_point(aes(x=china_master_heat$Centroid.X,
                 y=china_master_heat$Centroid.Y),
             data=china_master_heat, col="red", size=1) +
  geom_density2d(data=china_master_heat,
```

```

aes(x = china_master_heat$Centroid.X,
    y = china_master_heat$Centroid.Y,
    size = 0.3) +
stat_density2d(data=china_master_heat,
    aes(x = china_master_heat$Centroid.X,
        y = china_master_heat$Centroid.Y,
        fill = ..level..,
        alpha = ..level..),
    size = 0.01, geom = "polygon") +
scale_fill_gradient(low = "green", high = "red") +
scale_alpha(range = c(0, 0.3), guide = FALSE)

```

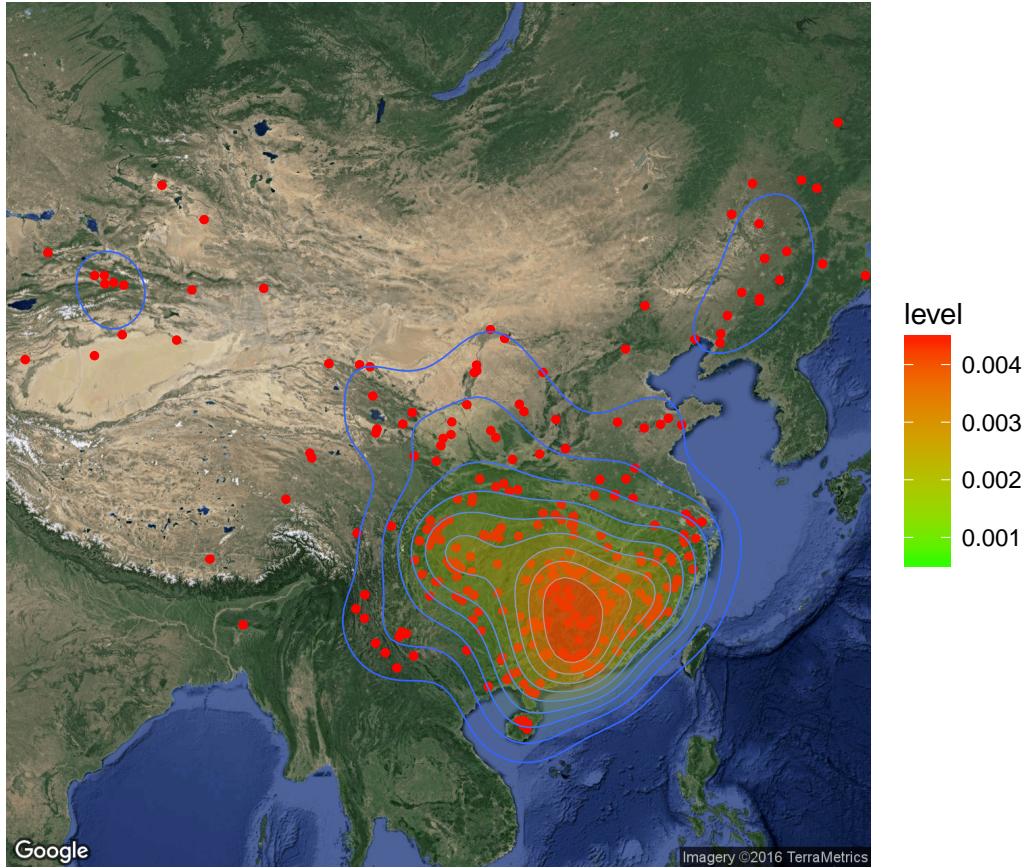
```

## Warning: Removed 5 rows containing non-finite values (stat_density2d).

## Warning: Removed 5 rows containing non-finite values (stat_density2d).

## Warning: Removed 5 rows containing missing values (geom_point).

```



```

##### information of India
## getting info of India
india_master = master_f[master_f$Country == "India", ]

## getting map of India
map_india <- get_map(location = "India", maptype = "satellite", zoom = 4)

```

```

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=India&zoom=4&size=640x640&scale=1
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=India&sensor=false

## Showing which area had floods in the past
india_master_heat = data.frame(Centroid.X=as.numeric(india_master$Centroid.X),
                                Centroid.Y=as.numeric(india_master$Centroid.Y),
                                Total.floods.M.4=india_master$Total.floods.M.4)

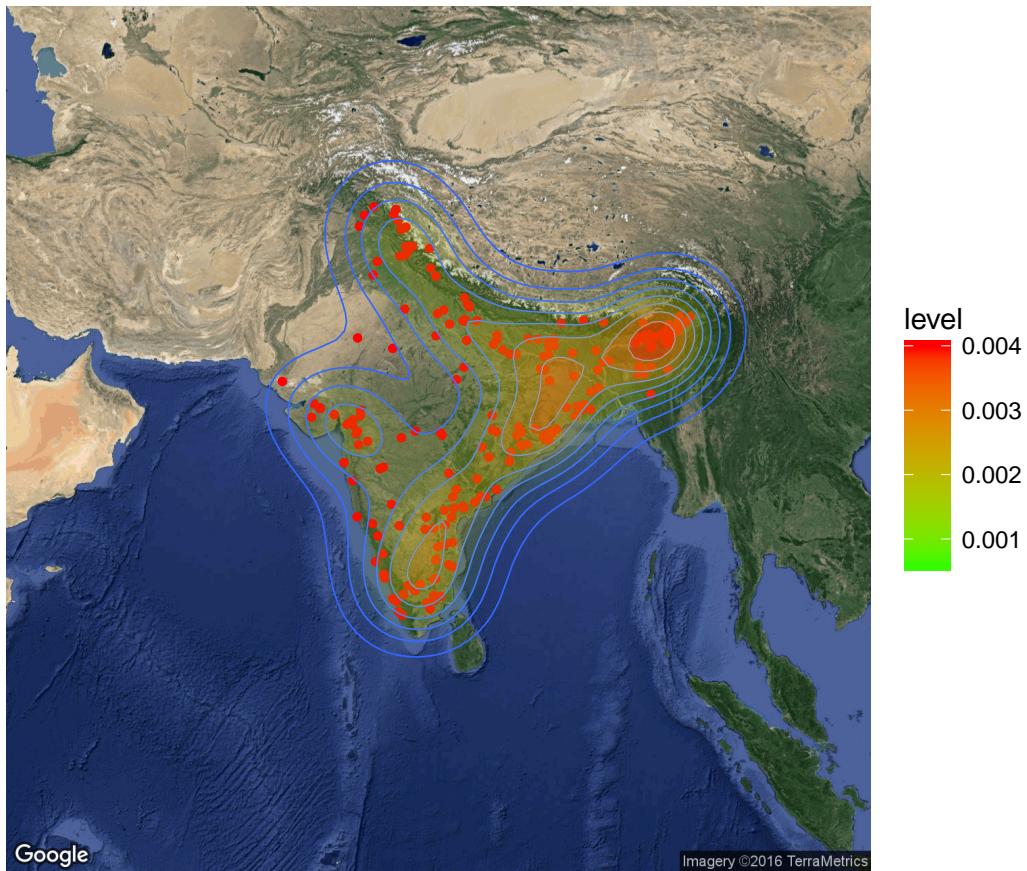
ggmap(map_india, extent = "device") +
  geom_point(aes(x=india_master_heat$Centroid.X,
                 y=india_master_heat$Centroid.Y),
             data=india_master_heat, col="red", size=1) +
  geom_density2d(data=india_master_heat,
                 aes(x = india_master_heat$Centroid.X,
                     y = india_master_heat$Centroid.Y,
                     size = 0.3) +
  stat_density2d(data=india_master_heat,
                 aes(x = india_master_heat$Centroid.X,
                     y = india_master_heat$Centroid.Y,
                     fill = ..level..,
                     alpha = ..level..),
                 size = 0.01, geom = "polygon") +
  scale_fill_gradient(low = "green", high = "red") +
  scale_alpha(range = c(0, 0.3), guide = FALSE)

## Warning: Removed 1 rows containing non-finite values (stat_density2d).

## Warning: Removed 1 rows containing non-finite values (stat_density2d).

## Warning: Removed 1 rows containing missing values (geom_point).

```



```
#### NOAA
```

```
library(RNetCDF)
```

```
## Warning: package 'RNetCDF' was built under R version 3.2.3
```

```
noaa = open.nc('NOAA_Daily_phi_500mb.nc')
data = read.nc(noaa)
xlon = data$X
ylat = rev(data$Y)
z = data$phi[,,1]

### extracting related information
# Register=4267
target = master_f[master_f$Register.== 4267, ]
target = target[1, ]
begin = as.Date("01-01-1948", format = "%d-%m-%Y")
flood_begin = difftime(as.Date("27-06-2015", format = "%d-%m-%Y"), begin)
flood_end = difftime(as.Date("29-06-2015", format = "%d-%m-%Y"), begin)
flood_begin = as.numeric(flood_begin)
flood_end = as.numeric(flood_end)
tmp_X = floor(as.numeric( target$Centroid.X ))
tmp_Y = floor(as.numeric( target$Centroid.Y ))

# focus on certian area
```

```

xlon = xlon - 180
tmp_xlon_range = (xlon > tmp_X - 20) == (xlon < tmp_X + 20)
tmp_ylat_range = (ylat > tmp_Y - 10) == (ylat < tmp_Y + 10)
tmp_xlon = xlon[tmp_xlon_range]
tmp_ylat = ylat[tmp_ylat_range]

tmp_time = data$phi[tmp_xlon_range, tmp_ylat_range, flood_begin:flood_end]

# for loop for tmp_ylat
# first day
for(i in 1 : dim(tmp_ylat)) {
  if (i == 1) {
    tmp_X = data.frame(tmp_xlon)
    tmp_Y = data.frame(rep(tmp_ylat[i], dim(tmp_xlon)))
    tmp_Z = data.frame(tmp_time[, i, 1])
  } else {
    tmp_X = rbind(tmp_X, data.frame(tmp_xlon))
    tmp_Y = rbind(tmp_Y, data.frame(rep(tmp_ylat[i], dim(tmp_xlon))))
    tmp_Z = rbind(tmp_Z, data.frame(tmp_time[, i, 1]))
  }
}

# -----
# start copy from stat-density-2d.R

stat_density_2d <- function(mapping = NULL, data = NULL, geom = "density_2d",
                               position = "identity", contour = TRUE,
                               n = 100, h = NULL, na.rm = FALSE, bins=0,
                               show.legend = NA, inherit.aes = TRUE, ...) {
  layer(
    data = data,
    mapping = mapping,
    stat = StatDensity2d,
    geom = geom,
    position = position,
    show.legend = show.legend,
    inherit.aes = inherit.aes,
    params = list(
      na.rm = na.rm,
      contour = contour,
      n = n,
      bins=bins,
      ...
    )
  )
}

stat_density2d <- stat_density_2d

StatDensity2d <-
  ggproto("StatDensity2d", Stat,

```

```

default_aes = aes(colour = "#3366FF", size = 0.5),
required_aes = c("x", "y"),

compute_group = function(data, scales, na.rm = FALSE, h = NULL,
                        contour = TRUE, n = 100,bins=0) {
  if (is.null(h)) {
    h <- c(MASS:::bandwidth.nrd(data$x), MASS:::bandwidth.nrd(data$y))
  }

  dens <- MASS:::kde2d(
    data$x, data$y, h = h, n = n,
    lims = c(scales$x$dimension(), scales$y$dimension())
  )
  df <- data.frame(expand.grid(x = dens$x, y = dens$y), z = as.vector(dens$z))
  df$group <- data$group[1]

  if (contour) {
    # StatContour$compute_panel(df, scales,bins=bins,...) # bad dots...
    if (bins>0){
      StatContour$compute_panel(df, scales,bins)
    } else {
      StatContour$compute_panel(df, scales)
    }
  } else {
    names(df) <- c("x", "y", "density", "group")
    df$level <- 1
    df$piece <- 1
    df
  }
}

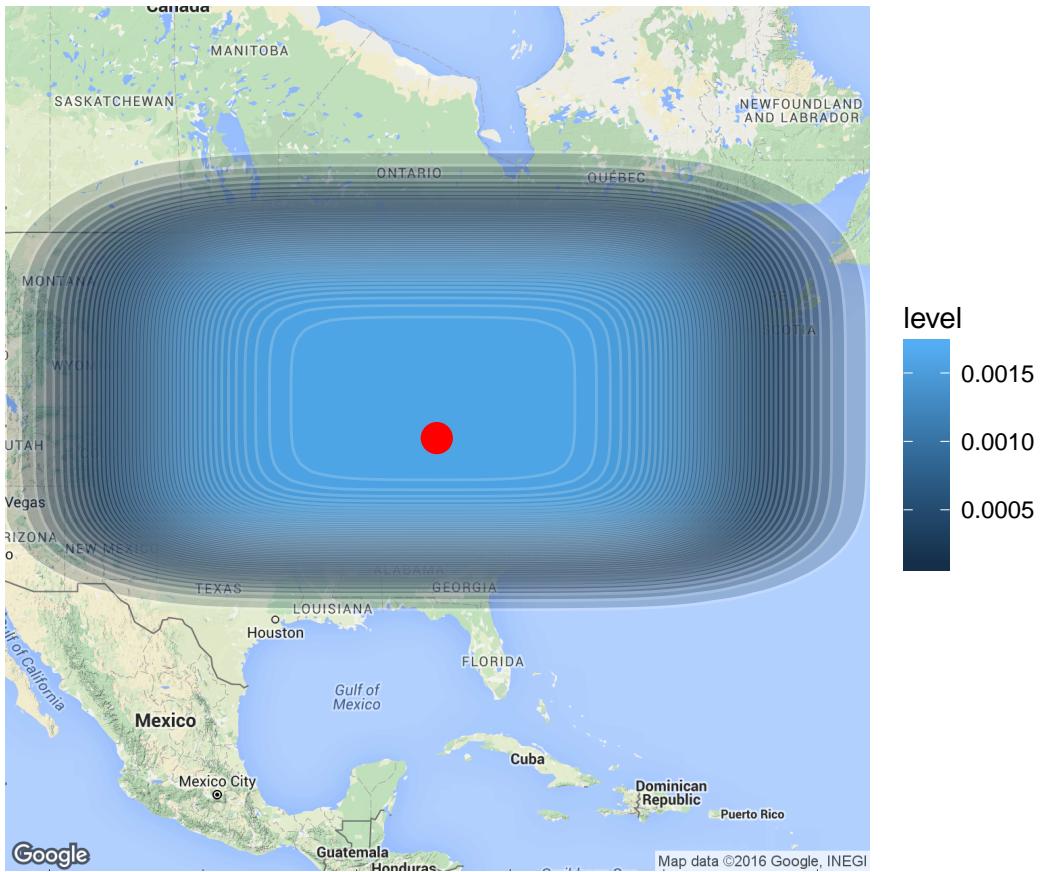
# end copy from stat-density-2d.R
# -----
map_4267 <- get_googlemap(location = "usa", center=c(as.numeric(target$Centroid.X),
                                                       as.numeric(target$Centroid.Y)),
                           zoom=4)

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=39.3942,-84.7486&zoom=4&size=640

ggmap(map_4267, extent = "device") +
  stat_density2d(
    aes(x = tmp_X$tmp_xlon,
        y = tmp_Y$rep.tmp_ylat.i...dim.tmp_xlon..,
        fill = ..level..),
    alpha = 0.2,
    bins = 50,
    data = data.frame(tmp_Z$tmp_time...i...1.),
    geom = "polygon") +
  geom_point(aes(x=as.numeric(target$Centroid.X),
                 y=as.numeric(target$Centroid.Y)),

```

```
data=target, col="red", size=5)
```



```
## stat bin
tmp_category = cut(tmp_Z$tmp_time...i...1., 10)

ggmap(map_4267, extent = "device") +
  stat_bin2d(data=data.frame(tmp_category),
             size = 3,
             alpha = 1/2,
             aes(x = tmp_X$tmp_xlon,
                 y = tmp_Y$rep.tmp_ylat.i...dim.tmp_xlon..,
                 colour = tmp_category,
                 fill = tmp_category ) ) +
  geom_point(aes(x=as.numeric(target$Centroid.X),
                 y=as.numeric(target$Centroid.Y)),
             data=target, col="red", size=5)
```

